



Oracle

1Z0-816 Exam

Oracle Java SE 11 Programmer II Exam

Thank you for Downloading 1Z0-816 exam PDF Demo

You can Buy Latest 1Z0-816 Full Version Download

<https://www.certkillers.net/Exam/1Z0-816>

<https://www.certkillers.net>

Version: 4.0

Question: 1

Given the code fragment:

```
Path currentFile = Paths.get("/scratch/exam/temp.txt");
```

```
Path outputFile = Paths.get("/scratch/exam/new.txt");
```

```
Path directory = Paths.get("/scratch/");
```

```
Files.copy(currentFile, outputFile);
```

```
Files.copy(outputFile, directory);
```

```
Files.delete (outputFile);
```

The /scratch/exam/temp.txt file exists. The /scratch/exam/new.txt and /scratch/new.txt files do not exist.

What is the result?

- A. /scratch/exam/new.txt and /scratch/new.txt are deleted.
- B. The program throws a FileAlreadyExistsException.
- C. The program throws a NoSuchFileException.
- D. A copy of /scratch/exam/new.txt exists in the /scratch directory and /scratch/exam/new.txt is deleted.

Answer: C

Explanation:

```
27 public class Main {  
28     public static void main(String[] args) {  
29         Path currentFile = Paths.get("/scratch/exam/temp.txt");  
30         Path outputFile = Paths.get("/scratch/exam/new.txt");  
31         Path directory = Paths.get("/scratch/");  
32  
33         Files.copy(currentFile, outputFile);  
34         Files.copy(outputFile, directory);  
35         Files.delete (outputFile);  
36     }  
37 }  
38
```

Question: 2

Which two are functional interfaces? (Choose two.)

- A. `@FunctionalInterface`
`interface MyRunnable {`
 `public void run();`
`}`
- B. `@FunctionalInterface`
`interface MyRunnable {`
 `public void run();`
 `public void call();`
`}`
- C. `interface MyRunnable {`
 `public default void run() {}`
 `public void run(String s);`
`}`
- D. `@FunctionalInterface`
`interface MyRunnable {`
`}`
- E. `interface MyRunnable {`
 `@FunctionalInterface`
 `public void run();`
`}`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: CE

Reference:

<http://tutorials.jenkov.com/java-functional-programming/functional-interfaces>

Question: 3

Given the declaration:

```
@interface Resource {  
    String name();  
    int priority() default 0;  
}
```

Examine this code fragment:

```
/* Loc1 */ class ProcessOrders { ... }
```

Which two annotations may be applied at Loc1 in the code fragment? (Choose two.)

- A. @Resource(priority=100)
- B. @Resource(priority=0)
- C. @Resource(name="Customer1", priority=100)
- D. @Resource(name="Customer1")
- E. @Resource

Answer: AB

Question: 4

Given:

```
interface MyInterface1 {  
    public int method() throws Exception;  
    private void pMethod() { /* an implementation of pMethod */ }  
}  
interface MyInterface2 {  
    public static void sMethod() { /* an implementation of sMethod */ }  
    public boolean equals();  
}  
interface MyInterface3 {  
    public void method();  
    public void method(String str);  
}  
interface MyInterface4 {  
    public void dMethod() { /* an implementation of dMethod */ }  
    public void method();  
}  
interface MyInterface5 {  
    public static void sMethod();  
    public void method(String str);  
}
```

Which two interfaces can be used in lambda expressions? (Choose two.)

- A. MyInterface1
- B. MyInterface3
- C. MyInterface5
- D. MyInterface2
- E. MyInterface4

Answer: CD

Reference:

<https://dzone.com/articles/functional-interface-and-lambda-expression>

Question: 5

Given this enum declaration:

```
1. enum Alphabet {  
2.     A, B, C  
3.  
4. }
```

Examine this code:

```
System.out.println(Alphabet.getFirstLetter());
```

What code should be written at line 3 to make this code print A?

- A. final String getFirstLetter() { return A.toString(); }
- B. static String getFirstLetter() { return Alphabet.values()[1].toString(); }
- C. static String getFirstLetter() { return A.toString(); }
- D. String getFirstLetter() { return A.toString(); }

Answer: C

Question: 6

Given these two classes:

```
public class Resource {  
    public Worker owner;  
    public synchronized boolean claim(Worker worker) {  
        if (owner == null) {  
            owner = worker;  
            return true;  
        }  
        else return false;  
    }  
    public synchronized void release() {  
        owner = null;  
    }  
}
```

```
public class Worker {  
    public synchronized void work(Resource... resources) {  
        for (int i = 0; i < 10; i++) {  
            while (!resources[0].claim(this)) { }  
            while (!resources[1].claim(this)) { }  
            // do work with resource  
            resources[1].release();  
            resources[0].release();  
        }  
    }  
}
```

And given this fragment:

```
Worker w1 = new Worker();  
Worker w2 = new Worker();  
Resource r1 = new Resource();  
Resource r2 = new Resource();  
new Thread( () -> {  
    w1.work(r1, r2);  
} ).start();  
new Thread( () -> {  
    w2.work(r2, r1);  
} ).start();
```

Which describes the fragment?

- A. It throws `IllegalMonitorStateException`.
- B. It is subject to deadlock.
- C. It is subject to livelock.
- D. The code does not compile.

Answer: D

Question: 7

Given:

```
public interface TestInterface {  
    default void samplingProbeProcedure() {  
        probeProcedure();  
        System.out.println("Collect Sample");  
        System.out.println("Leave Asteroid");  
        System.out.println("Dock with Main Craft");  
    }  
    default void explosionProbeProcedure() {  
        probeProcedure();  
        System.out.println("Explode")  
    }  
}
```

Examine these requirements:

- Eliminate code duplication.
 - Keep constant the number of methods other classes may implement from this interface.
- Which method can be added to meet these requirements?

- A. `private default void probeProcedure() {
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`
- B. `static void probeProcedure() {
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`
- C. `private void probeProcedure() {
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`
- D. `default void probeProcedure() {
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`

- A. Option A
B. Option B
C. Option C
D. Option D

Answer: B

Question: 8

Given:

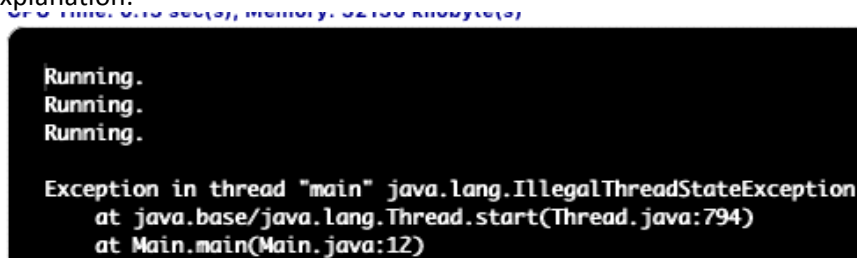
```
public class Main {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(new MyThread());  
        Thread t2 = new Thread(new MyThread());  
        Thread t3 = new Thread(new MyThread());  
  
        t1.start();  
        t2.run();  
        t3.start();  
  
        t1.start();  
    }  
}  
class MyThread implements Runnable {  
    public void run() {  
        System.out.println("Running.");  
    }  
}
```

Which one is correct?

- A. An `IllegalThreadStateException` is thrown at run time.
- B. Three threads are created.
- C. The compilation fails.
- D. Four threads are created.

Answer: A

Explanation:



```
Running.  
Running.  
Running.  
  
Exception in thread "main" java.lang.IllegalThreadStateException  
    at java.base/java.lang.Thread.start(Thread.java:794)  
    at Main.main(Main.java:12)
```

Question: 9

Which code fragment does a service use to load the service provider with a Print interface?

- A. `private Print print = com.service.Provider.getInstance();`
- B. `private java.util.ServiceLoader<Print> loader = ServiceLoader.load (Print.class);`
- C. `private java.util.ServiceLoader<Print> loader = new java.util.ServiceLoader<> ();`
- D. `private Print print = new com.service.Provider.PrintImpl();`

Answer: B

Reference:

<https://docs.oracle.com/javase/8/docs/api/?java/util/ServiceLoader>

Question: 10

Examine these module declarations:

```
module ServiceAPI {  
    exports com.example.api;  
}  
  
module ServiceProvider {  
    requires ServiceAPI;  
    provides com.example.api with com.myimpl.Impl;  
}  
  
module Consumer {  
    requires ServiceAPI;  
    uses com.example.api;  
}
```

Which two statements are correct? (Choose two.)

- A. The ServiceProvider module is the only module that, at run time, can provide the com.example.api API.
- B. The placement of the com.example.api API in a separate module, ServiceAPI, makes it easy to install multiple provider modules.
- C. The Consumer module should require the ServiceProvider module.
- D. The ServiceProvider module should export the com.myimpl package.
- E. The ServiceProvider module does not know the identity of a module (such as Consumer) that uses the com.example.api API.

Answer: AC

CertKillers.net

Thank You for trying 1Z0-816 PDF Demo

To Buy Latest 1Z0-816 Full Version Download visit link below

<https://www.certkillers.net/Exam/1Z0-816>

Start Your 1Z0-816 Preparation

[Limited Time Offer] Use Coupon “CKNET” for Further discount on your purchase. Test your 1Z0-816 preparation with actual exam questions.

<https://www.certkillers.net>